



APPLE -1

OPERATION MANUAL

APPLE COMPUTER COMPANY
770 Welch Road
Palo Alto, Calif. 94304

SPECIFICATIONS

MICROPROCESSOR:	MOS TECHNOLOGY 6502
Microprocessor Clock Frequency:	1.023 MHz
Effective Cycle Frequency: (Including Refresh Waits)	0.960 MHz
VIDEO OUTPUT:	Composite positive video, 75 ohms, level adjustable between zero and +5Vpp.
Line Rate:	15734 Hz
Frame Rate:	60.05 Hz
Format:	40 characters/line, 24 lines; with automatic scrolling
Display Memory:	Dynamic shift registers (1K x 7)
Character Matrix:	5 x 7
RAM MEMORY:	16-pin, 4K Dynamic, type 4096 (2104)
On-board RAM Capacity:	8K bytes (4K supplied)
POWER SUPPLIES:	+5 Volts @ 3 amps, +/- 12 Volts @0.5 amps, and -5 Volts @ 0.5 amps
Input Power Requirements:	8 to 10 Volts AC (RMS) @ 3 amps, 26 to 28 Volts AC (RMS) Center-Tapped, 1A.
Recommended Transformers:	Stancor # P-8380 or Triad F31-X Stancor # P-8667 or Triad F40-X

APPLE COMPUTER COMPANY
770 Welch Road, Suite 154
Palo Alto, California 94304
Phone: (415) 326-4248

INTRODUCTION

The Apple Computer is a complete microprocessor system, consisting of a Mos Technology 6502 microprocessor and support hardware, integral video display electronics, dynamic memory and refresh hardware, and fully regulated power supplies. It contains resident system monitor software, enabling the user, via the keyboard and display, to write, examine, debug, and run programs efficiently; thus being an educational tool for the learning of microprocessor programming, and an aid in the development of software.

The integral video display section and the keyboard interface renders unnecessary the need for an external teletype. The display section contains its own memory, leaving all of RAM for user programs, and the output format is 40 characters/line, 24 lines/page, with auto scrolling. Almost any ASCII encoded keyboard will interface directly with the Apple system.

The board has sockets for upto 8K bytes of the 16 pin, 4K type, RAM, and the system is fully expandable to 65K via the edge connector. The system uses dynamic memory (4K bytes sup-

plied), although static memory may also be used. All refreshing of dynamic memory, including all "off-board" expansion memory, is done automatically. The entire system timing, including the microprocessor clock and all video signals, originates in a single crystal oscillator.

Further, the printed circuit board contains a "breadboard area", in which the user can add additional "on-board" hardware (for example, extra PIA's, ACIA's, EROM's, and so on).

This manual is divided into three Sections:

Section I GETTING THE SYSTEM RUNNING.

Section II USING THE SYSTEM MONITOR.

(listing included)

Section III EXPANDING THE SYSTEM.

Please read Section I thoroughly, before attempting to "power-up" your system, and study Section III carefully before attempting to expand your system. In addition to this manual, Apple "Tech Notes" are available which contain examples of expansion hardware and techniques.

SECTION I GETTING THE SYSTEM RUNNING

The Apple Computer is fully assembled, tested, and burned in. The only external devices necessary for operation of the system are: An ASCII encoded keyboard, a video display monitor, and AC power sources of 8 to 10 Volts (RMS) @3 amps and 28Volts (RMS) @1 amp. The following three articles describe the attachment of these devices in detail.

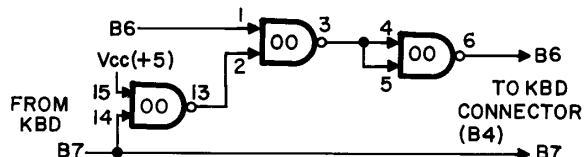
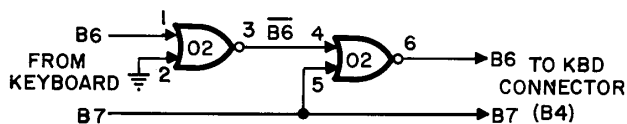
Keyboard:

Any ASCII encoded keyboard, with positive DATA outputs, interfaces directly with the Apple system via a "DIP" connector. If your keyboard has negative logic DATA outputs (rare), you can install inverters (7404) in the breadboard area. The strobe can be either positive or negative, of long or short duration. The "DIP" keyboard connector (B4) has inputs for seven DATA lines, one

STROBE line, and two normally-open pushbutton switches, used for RESET (enter monitor), and CLEAR SCREEN (see schematic diagram, sheet 3 of 3, for exact circuitry). This keyboard connector also supplies three voltages, (+5V, +12V, and -12V) of which one or more may be necessary to operate the keyboard. Pin 15 of the keyboard connector (B4) must be tied to +5V (pin 16) for normal operation.

NOTE: The system monitor accepts only upper-case alpha (A-F, R).

It is therefore convenient, though it's not essential, to have a keyboard equipped with upper-case alpha lock (usually in the electronics). Either of the following suggested circuits may be used to provide alpha lock capability, if needed, and can be built in the breadboard area.



Display:

The Apple Computer outputs a composite video signal (composite of sync and video information) which can be applied to any standard raster-scan type video display monitor. The output level is adjustable with the potentiometer located near the video output Molex connector, J2. The additional two outside pins on the Molex connector supply +5 and +12 volts, to be used in future Apple accessories. The composite video signal can also be modulated at the proper RF frequency, with an inexpensive commercially available device, and applied to the antenna terminals of a home television receiver. Since the character format is 40 characters/line, all television receivers will have the necessary bandwidth to display the entire 40 characters. Two large manufacturers of video display monitors, which connect directly with the Apple Computer, are Motorola and Ball. The mating four-pin Molex connector is provided.

AC Power Sources:

Two incoming AC power sources are required for operation: 8 to 10 VAC (RMS) at 3 amps, and 28 VAC (RMS) Center-Tapped at 1 amp. These AC supplies enter the system at the Molex connector, J1. The 8 to 10 volts AC provides the raw AC for the +5 volt supply, while the 28 VCT supplies the raw AC for the +12 and -12 volt supplies, and the -5V supply is derived from the -12V regulated output.

The board, as supplied, requires no more than 1.5 amps DC from the +5V supply, while the regulator is capable of supplying 3 amps. The remaining 1.5 amps DC from the +5V supply is available for user hardware expansion (provided suitable transformer ratings are employed).

A suitable source of the raw AC voltages required, are two commercially available transformers; Stancor P/N P-8380 or equivalent (8 to 10 volts at 3 amps), and Stancor P/N P-8667 or

equivalent (28VCT at 1 amp). Simply wire the secondaries to the mating six-pin Molex connector supplied, and wire the primaries in parallel, as shown in the schematic diagram (power supply section, Dwg.No. 00101, sheet 3 of 3).

TEST PROGRAM

After attaching the keyboard, display, and AC power sources, you can try a simple program to test if your system and the attachments are functioning together properly. While it does not test many possible areas of the microprocessor system, the test program will test for the correct attachment of the keyboard, display, and power supplies.

FIRST:

Hit the RESET button to enter the system monitor. A backslash should be displayed, and the cursor should drop to the next line.

SECOND:

Type- \emptyset : A9 b \emptyset b AA b 2 \emptyset b EF b FF b
E8 b 8A b 4C b 2 b \emptyset (RET)
(\emptyset is a zero, NOT an alpha "O"; b means blank or space; and (RET) hit the "return" key on the keyboard)

THIRD:

Type- \emptyset . A (RET)
(This should print out, on the display, the program you have just entered.)

FOURTH:

Type- R (RET)
(R means run the program.)

THE PROGRAM SHOULD THEN PRINT OUT ON THE DISPLAY A CONTINUOUS STREAM OF ASCII CHARACTERS. TO STOP THE PROGRAM AND RETURN TO THE SYSTEM MONITOR, HIT THE "RESET" BUTTON. TO RUN AGAIN, TYPE : R (RET).

SECTION II USING THE SYSTEM MONITOR

The Hex Monitor is a PROM program in locations FF00 to FFFF (hex) which uses the keyboard and display to perform the front panel functions of examining memory, and running programs. The monitor program is entered by hitting (RESET), which displays backslash-return. A backslash alone (cursor remains on same line as backslash) indicates bad page 0 RAM.

Commands are typed on a "line-at-a-time" basis with editing. Each line may consist of any number of commands (up to 128 characters). None are executed until (RETURN) is typed. The (SHIFT-0) (backarrow) backspaces and echos an underline. The (ESC) cnacels a line and echos backslash-return.

One or more hexadecimal digits (0-9, A-F) are used for address and data values. Addresses use the four least significant digits of a group, and data values, the two least significant digits. The following examples illustrate the variety of acceptable commands:

1. Opening a location (examining the contents of a single address).

```
USER TYPES/ 4F (RET)
MONITOR TYPES/ 004F: 0F (contents
                of 4F)
```

2. Examining a block; from the last examined location, to a specified one.

```
USER TYPES/ .5A (RET)
MONITOR TYPES/
0050: 00 01 02 03 04 05 06 07
0058: 08 09 0A
```

Note: 4F is still considered the most recently opened location.

3. Combining examples 1 and 2 to print a block of memory in a single command.

```
USER TYPES/ 4F.5A (RET)
MONITOR TYPES/
0050: 00 01 02 03 04 05 06 07
0058: 08 09 0A
```

Note: Only the first location of the block (4F) is considered "opened".

4. Examining several individual locations at once.

```
USER TYPES/ 4F b 52 b 56 (RET)
MONITOR TYPES/ 004F: 0F
                0052: 02
                0056: 06
```

Note: 56 is considered the most recently "opened" location. The "b" is a blank or comma, and is a delimiter for separation purposes only. A string of delimiters has the same effect as a single one (bbb is as effective as b).

5. Examining several blocks of memory at once.

```
USER TYPES/ 4F.52 b 56 b 58.5A
                (RET)
MONITOR TYPES/ 004F: 0F
                0050: 00 01 02
                0056: 06
                0058: 08 09 0A
```

Note: 58 is considered the most recently "opened" location. Refer to example 2.

6. Examining successive blocks.

```
USER TYPES/ 4F.52 (RET)
MONITOR TYPES/ 004F: 0F
                0050: 00 01 02
                USER TYPES/ .55 (RET)
MONITOR TYPES/ 0053: 03 04 05
                USER TYPES/ .5A (RET)
MONITOR TYPES/ 0056: 06 07
                0058: 08 09 0A
```

7. Depositing data in a single location.

```
USER TYPES/ 30: A0 (RET)
MONITOR TYPES/ 0030: FF (prior
                contents)
```

Note: Location 30 is considered opened and now contains 30.

8. Depositing data in successive locations from that last used in a deposit command.

```
USER TYPES/ : A1 b A2 b A3 b A4
                b A5 (RET)
(This deposits A1 in location 31, A2 in 32,
and so on.)
```

9. Combining examples 7 and 8 in a single command.

```
USER TYPES/ 30: A0 b A1 b A2 b
                A3 b A4 b A5 (RET)
MONITOR TYPES/
0030: FF (prior contents of location 30)
```

10. Depositing data in successive locations with separate commands.

```
USER TYPES/ 30: A0 b A1 (RET)
MONITOR TYPES/ 0030: FF
                USER TYPES/ :A2 b A3 (RET)
                USER TYPES/ :A4 b A5 (RET)
```

NOTE: Capital letters enclosed in parenthesis represent single keystrokes.

Example: (RET) means hit the "return" key.

Note: A colon in a command means "start depositing data from the most recently deposited location, or if none, then from the most recently opened one.

11. Examining a block, then depositing into it.
 USER TYPES/ 30.35 (RET)
 MONITOR TYPES/
 0030: A0 A1 A2 A3 A4 A5 A6
 USER TYPES/
 :B0 b B1 b B2 b B3 b B4 b B5 (RET)

Note: New data deposited beginning at most recently opened location (30)

12. Run a program at a specified address.
 USER TYPES/ 10F0 R (RET)
 MONITOR TYPES/ 10F0: A9 (contents)

Note: The cursor is left immediately to the right of the "A9"; it is not returned to the next line.

13. Run at the most recently examined location.
 USER TYPES/ 10F0 (RET)
 MONITOR TYPES/ 10F0: A9
 USER TYPES/ R (RET)

14. Enter a program into memory and run it in one line.
 USER TYPES/
 40: A9 b 0 b 20 b EF b FF b 38 b 69 b
 0 b 4C b 40 b 0 R (RET)
 MONITOR TYPES/ 40: FF (prior contents of 40)

MONITOR TYPES/ 40: FF (prior contents of 40)

15. An "on line" error correction.
 USER TYPES/
 40: A1 b A2 b A3A4A5A6 b A7
 (data A6 will be loaded in location 42)
 USER TYPES/ 40506070: AA
 (data AA will be loaded in location 6070)
16. Useful routines in monitor which can be accessed by user programs.
 GETLINE: location FF1F:
 monitor entry point
 (jumping to FF1F will enter monitor and echo carriage return. You can then examine memory locations with the monitor.)

ECHO: location FFEF:
 prints one byte (ASCII)
 (data from "A" (accumulator), contents of "A" not disturbed. Example:
 20 b EF b FF (JRS ECHO)).

PRBYTE: location FFDC:
 prints one byte (HEX)
 (data from "A", contents of "A" disturbed.)

PRHEX: location FFE5:
 prints one hex digit
 (data from four least significant bits of "A", contents of "A" disturbed.)

NOTE: RAM locations 0024 to 002B are used as index pointers by the monitor, and are invalid for user use, when using monitor. Also, locations 0200 to 027F are used as input buffer storage, and are also invalid for user use when using the monitor.

6502 HEX MONITOR LISTING

FF00	D8	RESET	CLD	Clear decimal arithmetic mode.
FF01	58		CLI	
FF02	A0 7F		LDY #\$7F	Mask for DSP data direction register.
FF04	8C 12 D0		STY DSP	Set it up.
FF07	A9 A7		LDA #\$A7	KBD and DSP control register mask.
FF09	8D 11 D0		STA KBD CR	Enable interrupts, set CA1, CB1, for
FF0C	8D 13 D0		STA DSP CR	positive edge sense/output mode.
FF0F	C9 DF	NOTCR	CMP #\$DF	"←"?
FF11	F0 13		BEQ BACKSPACE	Yes.
FF13	C9 9B		CMP #\$9B	ESC?
FF15	F0 03		BEQ ESCAPE	Yes.
FF17	C8		INY	Advance text index.
FF18	10 0F		BPL NEXTCHAR	Auto ESC if >127.
FF1A	A9 DC	ESCAPE	LDA #\$DC	"\".
FF1C	20 EF FF		JSR ECHO	Output it.
FF1F	A9 8D	GETLINE	LDA #\$8D	CR.
FF21	20 EF FF		JSR ECHO	Output it.
FF24	A0 01		LDY #\$01	Initiallize text index.
FF26	88	BACKSPACE	DEY	Back up text index.
FF27	30 F6		BMI GETLINE	Beyond start of line, reinitialize.
FF29	AD 11 D0	NEXTCHAR	LDA KBD CR	Key ready?
FF2C	10 FB		BPL NEXTCHAR	Loop until ready.
FF2E	AD 10 D0		LDA KBD	Load character. B7 should be '1'.
FF31	99 00 02		STA IN, Y	Add to text buffer.
FF34	20 EF FF		JSR ECHO	Display character.
FF37	C9 8D		CMP #\$8D	CR?
FF39	D0 D4		BNE NOTCR	No.
FF3B	A0 FF		LDY #\$FF	Reset text index.
FF3D	A9 00		LDA #\$00	For XAM mode.
FF3F	AA		TAX	0→X.
FF40	0A	SETSTOR	ASL	Leaves \$7B if setting STOR mode.
FF41	85 2B	SETMODE	STA MODE	\$00 = XAM, \$7B = STOR, \$AE = BLOK XAM.
FF43	C8	BLSKIP	INY	Advance text index.
FF44	B9 00 02	NEXT ITEM	LDA IN, Y	Get character.
FF47	C9 8D		CMP #\$8D	CR?
FF49	F0 D4		BEQ GETLINE	Yes, done this line.
FF4B	C9 AE		CMP #\$AE	". "?
FF4D	90 F4		BCC BLSKIP	Skip delimiter.
FF4F	F0 F0		BEQ SETMODE	Set BLOCK XAM mode.
FF51	C9 BA		CMP #\$BA	": "?
FF53	F0 EB		BEQ SETSTOR	Yes, set STOR mode.
FF55	C9 D2		CMP #\$D2	"R"?
FF57	F0 3B		BEQ RUN	Yes, run user program.
FF59	86 28		STX L	\$00→L.
FF5B	86 29		STX H	and H.
FF5D	84 2A		STY YSAV	Save Y for comparison.
FF5F	B9 00 02	NEXTHEX	LDA IN, Y	Get character for hex test.
FF62	49 B0		EOR #\$B0	Map digits to \$0-9.
FF64	C9 0A		CMP #\$0A	Digit?
FF66	90 06		BCC DIG	Yes.
FF68	69 88		ADC #\$88	Map letter "A"- "F" to \$FA-FF.
FF6A	C9 FA		CMP #\$FA	Hex letter?
FF6C	90 11		BCC NOTHEX	No, character not hex.
FF6E	0A	DIG	ASL	
FF6F	0A		ASL	Hex digit to MSD of A.
FF70	0A		ASL	
FF71	0A		ASL	
FF72	A2 04		LDX #\$04	Shift count.
FF74	0A	HEXSHIFT	ASL	Hex digit left, MSB to carry.

6502 HEX MONITOR LISTING (continued)

FF75	26 28		ROL L	Rotate into LSD.
FF77	26 29		ROL H	Rotate into MSD's.
FF79	CA		DEX	Done 4 shifts?
FF7A	D0 F8		BNE HEXSHIFT	No, loop.
FF7C	C8		INY	Advance text index.
FF7D	D0 E0		BNE NEXTHEX	Always taken. Check next character for hex.
FF7F	C4 2A	NOTHEX	CPY YSAV	Check if L, H empty (no hex digits).
FF81	F0 97		BEQ ESCAPE	Yes, generate ESC sequence.
FF83	24 2B		BIT MODE	Test MODE byte.
FF85	50 10		BVC NOTSTOR	B6 = 0 for STOR, 1 for XAM and BLOCK XAM
FF87	A5 28		LDA L	LSD's of hex data.
FF89	81 26		STA (STL, X)	Store at current 'store index'.
FF8B	E6 26		INC STL	Increment store index.
FF8D	D0 B5		BNE NEXTITEM	Get next item. (no carry).
FF8F	E6 27		INC STH	Add carry to 'store index' high order.
FF91	4C 44 FF	TONEXTITEM	JMP NEXTITEM	Get next command item.
FF94	6C 24 00	RUN	JMP (XAML)	Run at current XAM index.
FF97	30 2B	NOTSTOR	BMI XAMNEXT	B7 = 0 for XAM, 1 for BLOCK XAM.
FF99	A2 02		LDX #02	Byte count.
FF9B	B5 27	SETADR	LDA L-1, X	Copy hex data to
FF9D	95 25		STA STL-1, X	'store index'.
FF9F	95 23		STA XAML-1, X	And to 'XAM index'.
FFA1	CA		DEX	Next of 2 bytes.
FFA2	D0 F7		BNE SETADR	Loop unless X = 0.
FFA4	D0 14	NXTPRNT	BNE PRDATA	NE means no address to print.
FFA6	A9 8D		LDA #8D	CR.
FFA8	20 EF FF		JSR ECHO	Output it.
FFAB	A5 25		LDA XAMH	'Examine index' high-order byte.
FFAD	20 DC FF		JSR PRBYTE	Output it in hex format.
FFB0	A5 24		LDA XAML	Low-order 'examine index' byte.
FFB2	20 DC FF		JSR PRBYTE	Output it in hex format.
FFB5	A9 BA		LDA #BA	":":.
FFB7	20 EF FF		JSR ECHO	Output it.
FFBA	A9 A0	PRDATA	LDA #A0	Blank.
FFBC	20 EF FF		JSR ECHO	Output it.
FFBF	A1 24		LDA (XAML, X)	Get data byte at 'examine index'.
FFC1	20 DC FF		JSR PRBYTE	Output it in hex format.
FFC4	86 2B	XAMNEXT	STX MODE	0 → MODE (XAM mode).
FFC7	A5 24		LDA XAML	
FFC8	C5 28		CMP L	Compare 'examine index' to hex data.
FFCA	A5 25		LDA XAMH	
FFCC	E5 29		SBC H	
FFCE	B0 C1		BCS TONEXTITEM	Not less, so no more data to output.
FFD0	E6 24		INC XAML	
FFD2	D0 02		BNE MOD8CHK	Increment 'examine index'.
FFD4	E6 25		INC XAMH	
FFD6	A5 24	MOD8CHK	LDA XAML	Check low-order 'examine index' byte
FFD8	29 07		AND #07	For MOD 8 = 0
FFDA	10 C8		BPL NXTPRNT	Always taken.
FFDC	48	PRBYTE	PHA	Save A for LSD.
FFDD	4A		LSR	
FFDE	4A		LSR	
FFDF	4A		LSR	MSD to LSD position.
FFE0	4A		LSR	
FFE1	20 E5 FF		JSR PRHEX	Output hex digit.
FFE4	68		PLA	Restore A.
FFE5	29 0F	PRHEX	AND #0F	Mask LSD for hex print.
FFE7	09 B0		ORA #B0	Add "0".
FFE9	C9 BA		CMP #BA	Digit?

6502 HEX MONITOR LISTING (continued)

FFEB	90 02	BCC ECHO	Yes, output it.
FFED	69 06	ADC #06	Add offset for letter.
FFEF	2C 12 D0	BIT DSP	DA bit (B7) cleared yet?
FFF2	30 FB	BMI ECHO	No, wait for display.
FFF4	8D 12 D0	STA DSP	Output character. Sets DA.
FFF7	60	RTS	Return.
FFF8	00 00 (unused)		
FFFA	00 0F (NMI)		
FFFC	00 FF (RESET)		
FFFE	00 00 (IRQ)		

HARDWARE NOTES

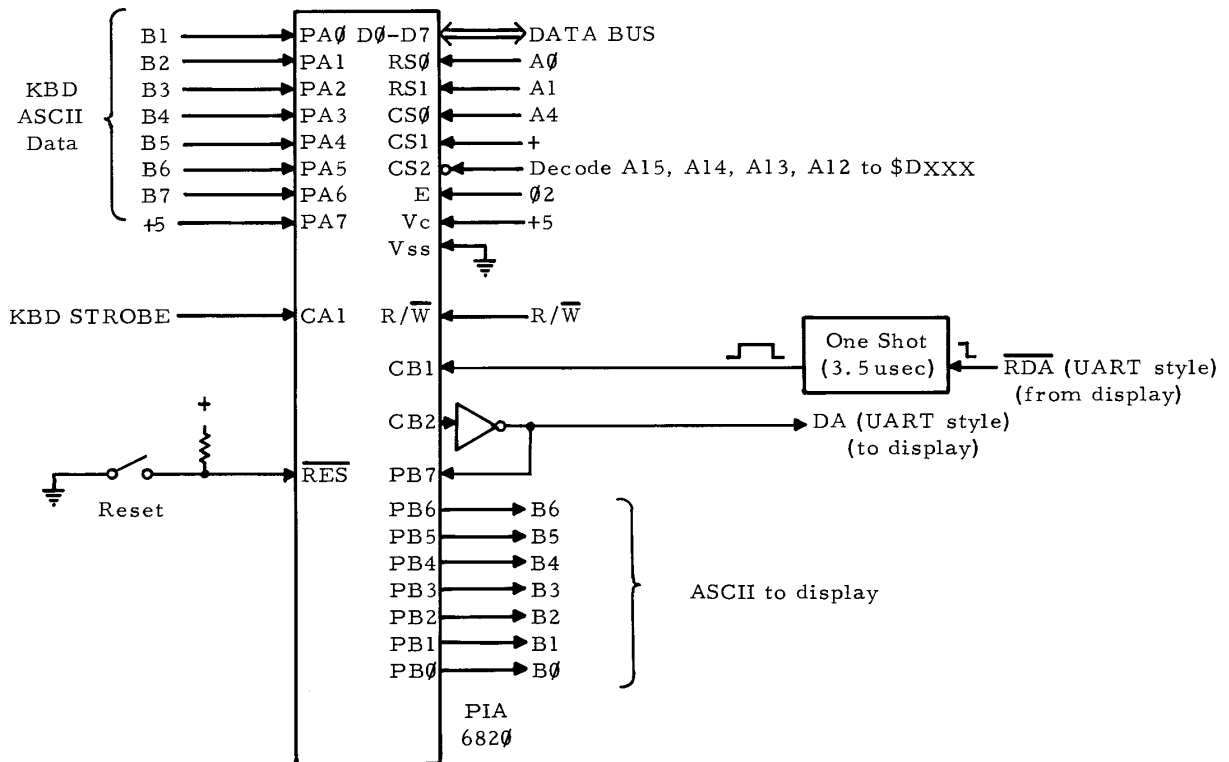
Page 0 Variables

XAML	24
XAMH	25
STL	26
STH	27
L	28
H	29
YSAV	2A
MODE	2B

Other Variables

IN	200-27F	} PIA
KBD	D010	
KBD CR	D011	
DSP	D012	
DSP CR	D013	

KBD/DSP Interface



SECTION III HOW TO EXPAND THE APPLE SYSTEM

The Apple system can be expanded to include more memory and IO devices, via a 44-pin edge connector. The system is fully expandable to 65K, with the entire data and address busses, clocks, control signals (i. e. IRQ, NMI, DMA, RDY, etc.), and power sources available at the connector. All address lines are TTL buffered, and data lines can drive ten equivalent capacitive loads (one TTL load and 130pf) without external buffers. All clock signals are TTL. The Apple system runs at approximately 1 MHz (see spec sheet) and is fully compatible with 6800/6500 style timing.

Three power sources are available at the edge connector: +5 volts regulated, and raw DC (approximately +/- 14V) for the +12V, -12V, and -5V supplies. If +12V, -12V, or -5V supplies are required, EXTERNAL REGULATORS MUST BE USED. An excess of 1.5 amps from the "on-board" regulated +5V supply is available for expansion (assuming suitable transformer ratings are employed). Exercise great care in the handling of the raw DC, as no short-circuit protection is provided.

REFRESH:

Four out of every 65 clock cycles is dedicated to memory refresh. At the start of a refresh cycle (150 ns after leading edge of $\phi 1$), \overline{RF} goes low, and remains low for one clock cycle. $\phi 2$ is inhibited during a refresh cycle, and the processor is held in $\phi 1$ (it's inactive state). Dynamic memories, which must clock during refresh cycles, should derive their clock from $\phi 0$, which is equivalent to $\phi 2$, except that it continues during a refresh cycle. Devices, such as PIA's, will not be affected by a refresh cycle, since they react to $\phi 2$ only. Refer to Apple "Tech Notes" for a variety of interfacing examples.

DMA:

The Apple system has full DMA capability. For DMA, the DMA control line tri-states the address buss, thus allowing external devices to control the buss. Consult MOS TECHNOLOGY 6502 Hardware Manual for details. (For DMA use, the solder jumper on the board, marked "DMA", must be broken.)

For the 6502 microprocessor, the RDY line is used to halt the processor for single stepping, or slow ROM applications. Refer to Apple "Tech Notes" for examples.

SOFTWARE CONSIDERATIONS:

The sequences listed below are the routines used to read the keyboard or output to the display.

Read Key from KBD:

```
LDA KBD CR (D011)
BPL
LDA KBD DATA (D010)
```

Output to Display:

```
BIT DSP (D012)
BPL
STA DSP (D012)
```

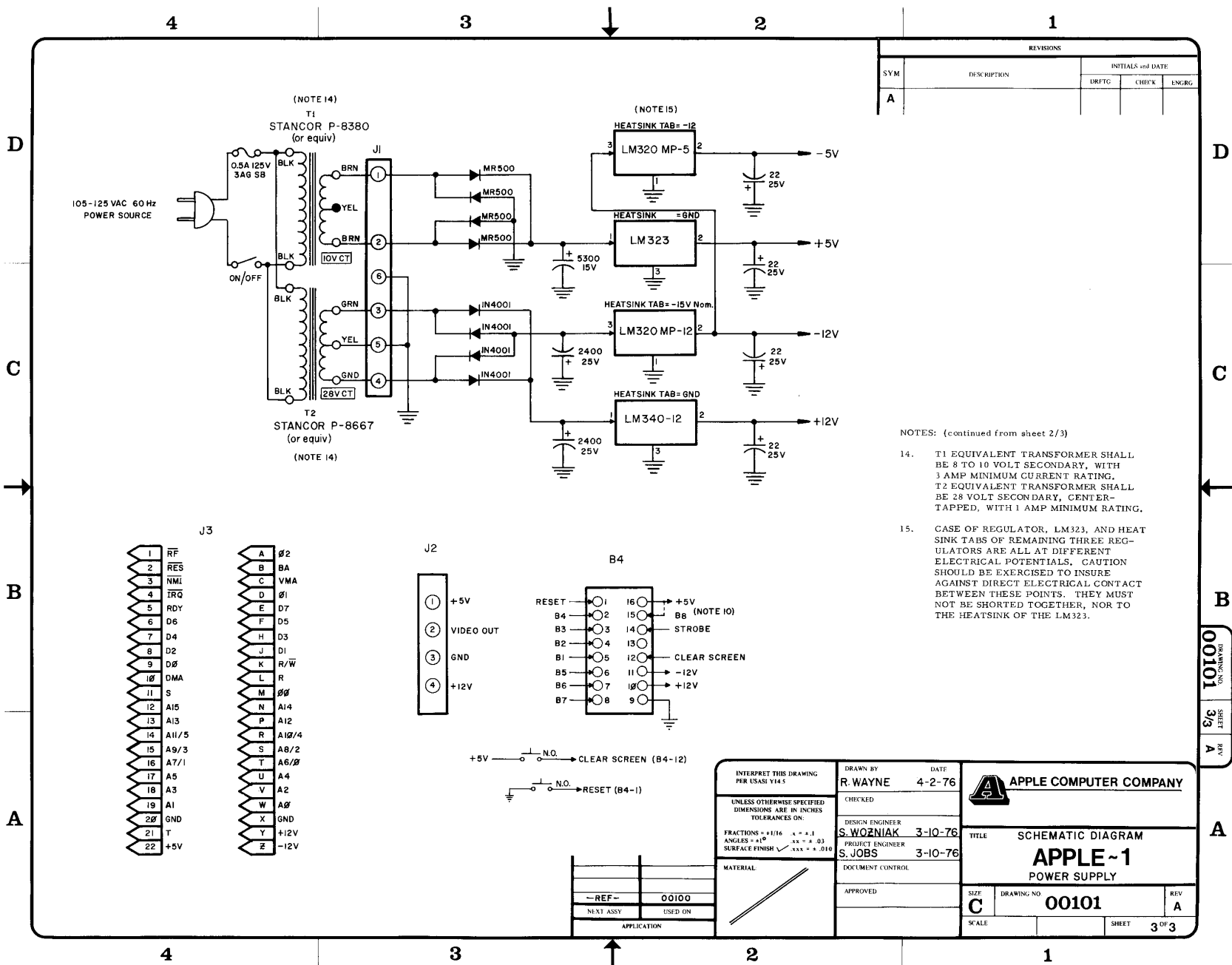
PIA Internal Registers:

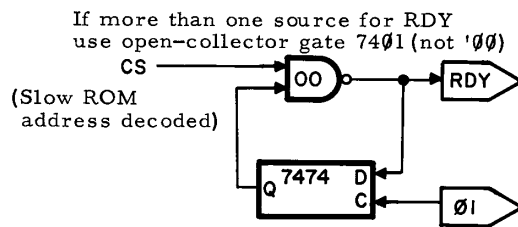
KBD Data D010
High order bit equals 1.

KBD Control Reg. D011
High order bit indicates "key ready".
Reading key clears flag. Rising edge of KBD sets flag.

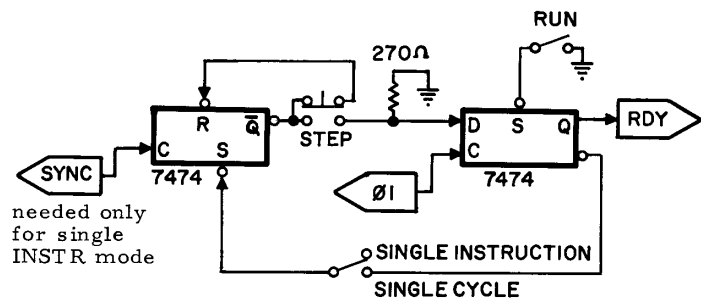
DSP DATA D012
Lower seven bits are data output,
high order bit is "display ready"
input (1 equals ready, 0 equals busy)

DSP Control Reg. D013





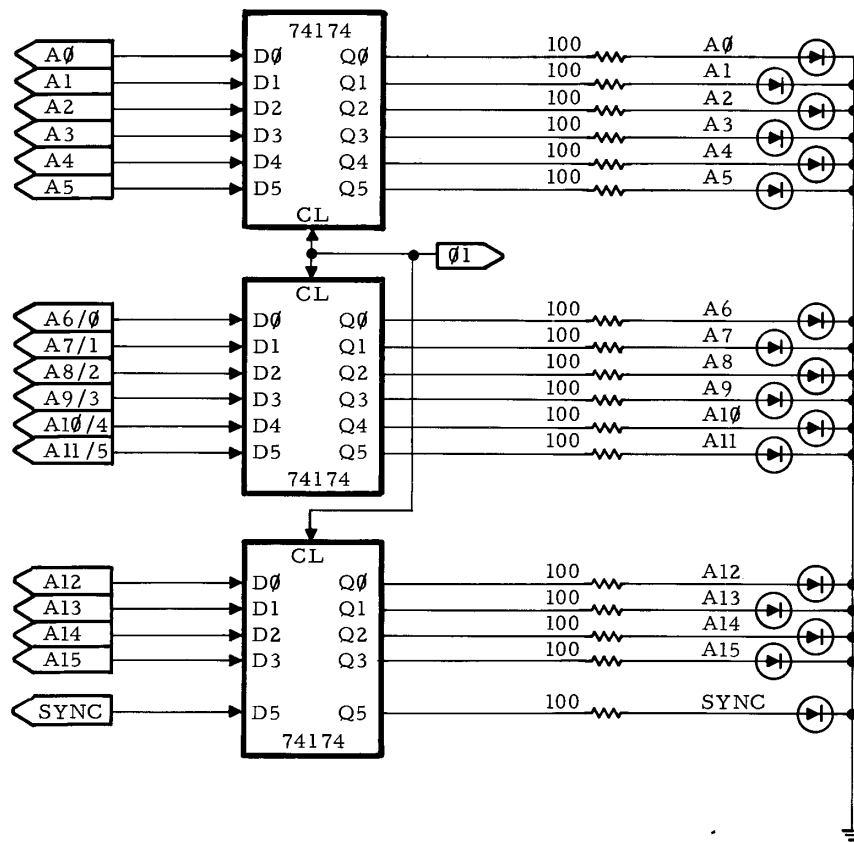
SLOW ROM



(NOTE: Features not needed may be omitted)

SINGLE STEP FOR 6502

ADDRESS DISPLAY





WARRANTY

The Apple Computer Company hereby warrants each of its products, and all components therein contained, to be free from defects in materials and/or workmanship for a period of thirty (30) days from date of purchase. In the event of the occurrence of malfunction, or other indication of failure attributable directly to faulty workmanship and/or material, then, upon return of the product to the Apple Computer Company, at 770 Welch Road, Palo Alto, California, 94304 (postage prepaid), the Apple Computer Company will, at its option, repair or replace said products or components thereof, to whatever extent Apple Computer Company shall deem necessary, to restore said product to proper operating condition. All such repairs or replacements shall be rendered by the Apple Computer Company, without charge to the customer.

The responsibility for the failure of any Apple Computer product, or component thereof, which, at the discretion of the Apple Computer Company, shall have resulted either directly or indirectly from accident, abuse, or misapplication of the product, shall be assumed by the customer, and the Apple Computer Company shall assume no liability as a consequence of such events under the terms of this warranty.

While every effort, on the part of Apple Computer Company, is made to provide clear and accurate technical instruction on the use, implementation, and application of its products, the Apple Computer Company shall assume no liability in events which may arise from the application of such technical instruction, nor shall the Apple Computer Company be held liable for the quality, interconnection, or application of peripheral products, which may have been recommended by Apple Computer Company, but which have not been supplied as part of the product.

This warranty contains and embodies the limits of responsibility of the Apple Computer Company, with regard to its products, and no other liability is expressed, implied, or should be assumed by the purchaser, and in no event shall the Apple Computer Company be held liable for the loss of time, effort, or transportation costs, nor for loss of potential profits or other consequential losses which might arise from the purchase, assembly, use, application, or subsequent sale of the products of Apple Computer Company, nor from any instructions and/or technical information thereto related.

